SENIOR DESIGN PROJECT REPORT

Automated Power Supply Test Unit

Submitted To

Professor Cooney

Professor Weissbach

Professor Lin

Electrical Engineering Technology Program

Engineering Technology Department

By

K. Mitchell Brauns

Yazed Alanazi

05/01/2020

EXECUTIVE SUMMARY

This project will be completed by the request of Professor Cooney. The goal of this senior design project is to create an automated power supply test unit. This automated power supply test unit is needed to reduce the time to test students' power supply projects. This report will cover the design process and testing process in depth. A replica of the project should easily be made using this documentation. The zero-crossing detector, triac control circuit, DC measurement circuit, and ripple voltage measurement circuit are all thoroughly documented. The full code for the Arduino Due and Nano is also included. Test results are shown for each step of the testing process. Overall, the project was a success. Small improvements could be made and will be covered in the recommendation portion.

Contents

Definitions

The following definitions, acronyms, and abbreviations are found in this document:

Table 1 - Definitions

Term	Definition
Power supply test unit	The *power supply test unit* is the final product that is to be built for this ECET 49000 project. It is the unit that will automatically test the students' power supplies.
Power supply	The *power supply* is referring to the ECET 15700 student's power supply.

Version History

Version	Change Made	Date
1.0	Document created	11/13/2019
1.5	DC & AC measurement Circuit added.	11/19/2019
2.0	AC voltage control circuit added. Input / output table added.	11/23/2019
2.5	DC and AC circuits revised with different resistor values and potentiometers added.	11/24/2019
3.0	Interface Design and function table included in document.	12/2/2019
3.5	Detailed software flowcharts included for all subroutines.	12/6/2019
4.0	Bill of materials added. Screen dimensions added to tables. Gantt chart added.	1/18/2020

Introduction

This project will be completed by the request of Professor Cooney. The goal of this senior design project is to create an automated power supply test unit. This automated power supply test unit is needed to reduce the time to test students' power supply projects. This will allow more time for Professor Cooney to help and guide students in their projects.

Project Scope

The scope of this project includes controlling the input voltage and load resistance that will be connected to the power supplies. It must also measure the power supply output voltage (DC & AC voltage), DC current out, and ensure that the test lasts 5 minutes. The results must then be recorded and stored on a computer via USB.

Out of Scope

Things considered to be out of scope for this project will be supplying the power to run the tester, building, helping to build, or troubleshooting power supplies.

Related Projects

The power supply projects in ECET 157 are indirectly related to this project. They are related for two reasons. One is that they partially control the project timeline. Our project must be done in time to test the power supplies. The other reason is we must design our project to test these power supplies. This means we must build our project to conform to their testing needs.

Assumptions

It is assumed that the ECET 15700 students will design their power supplies to the given specifications. This means that we should not have to design the tester to test for ranges outside of the 0-20V range. It can also be assumed that the students will not need to test for values above 1.5 amps.

Process Overview

(Current System)

The current system for testing the power supplies involves the professor manually testing all aspects of the system. This process involves several time-consuming steps. The first step is to set up the testing equipment, including the oscilloscope, DMM, load resistors, variac, and any data recording equipment. The professor must then connect the student's power supply to the testing equipment. After all equipment is set up and connected together, the professor must manually run the test. The testing process is started by running the power supply on the 15-ohm load and recording the AC ripple voltage, the DC voltage, and the DC current. This data must be recorded onto the data sheet manually. The professor must time the test for 5 minutes to ensure that the power supply does not fail. The power supply must also be tested with a lower AC input voltage, which is controlled by the variac. This is a time-consuming process that must be streamlined through automation.

(New Process)

The new process will automate as many of the steps as possible. The goal is to have the AC and DC voltage automatically measured, the loads automatically switched, and all data recorded without the input of a user. Doing this should save the instructor time.

Requirements

Customer Requirements

The power supply test unit must control the input voltage and load resistance that will be connected to the power supplies. It must also measure the power supply output voltage (DC & AC voltage), DC current out, and ensure that the test lasts 5 minutes. The results must then be recorded and stored on a computer via USB.

Engineering Specifications

1. Unit power supply test unit should test power supply at 1 amp.
2. The power supply test unit must test the power supply for exactly 5 minutes.
3. The AC input voltage must be varied automatically from 125 to 110 V RMS.
4. The power supply test unit output voltage must be within 1V rms of the specified output.
5. The power supply test unit must be accurate to 100mV when measuring DC voltage.

6. The power supply test unit must be accurate to 0.5mV when measuring AC ripple voltage.
7. Reverse polarity protection should be included for the power supply test unit to protect from the student connecting their power supply backwards.
8. User interface screen for instructions and status. Must display items at or above the size of 1cm.
9. USB connection to computer with autosave function to a pre-determined file. Must complete the data transfer in under two minutes.
10. The AC voltage measurement must cover the 1mV-3V range.
11. The power supply test unit must be able to run for a continuous time of 2 hours minimum.
12. The power supply test unit must be able to dissipate a minimum of 15 watts continuously for 2 hours.
13. The power supply test unit must be able to measure DC voltage within the 0-20V range.
14. The power supply test unit must have a built-in safety margin for the DC voltage it is measuring. The test unit should be able to safely handle up to a 50V DC input.

Hardware

Inputs: DC Voltage, AC voltage

Outputs: AC voltage (line voltage), USB out to computer

User Roles and Responsibilities

- The user must set up the power supply test unit. This includes plugging in the USB to the computer, opening the corresponding program, and plugging the tester into the provided wall outlet.
- The user must also connect the power supply that is to be tested to the test unit. This consists of connecting the positive and negative terminals to the corresponding locations.
- The power supply voltage out must also be manually controlled by the user. There will be an indication screen showing when to change the voltage, and to what level it should be changed.

System Wide Design Options

Microcontroller Options

Microcontroller	ADC Resolution (Number of bits)	Price	Number of Digital Inputs and outputs	Number of Analog Inputs	Programming Language
ESP8266	10 bits	$2 (sale)	11	1	C, C++
Arduino Mega	10 bits	$28	54	14	C, C++
Raspberry-Pi	16 bits (add on required)	$35	16		Java, C, Python
Arduino Uno	10 bits	$18	14	5	C, C++
Arduino Due	12 bits	$34	54	11	C, C++

Microcontroller Selection and Reasoning

The Arduino Due is the best microcontroller choice due to several factors. The large number of available pins, higher than normal ADC resolution, reasonable price, high number of analog pins, and ease of use due to the C/C++ programming language are all contributing factors that lead to the choice of this microcontroller.

AC Voltage Controller Options

Controller Type	Weight	Waveform Generated	Price	Footprint Size
Phase Control (Triac)	Light	Square Wave	$20-$50	Small
Variac	Heavy	Sine Wave	$30-$100	Large
"On-Off Control"	Light	Sine Wave	$20	Small

AC Voltage Controller Selection Reasoning

Triac phase control will be used to control the AC voltage supply to the student's power supply. This decision was made due to several factors. Triac phase

control allows for lighter components to be used, along with the voltage to be controlled by the microcontroller. This will also be the cheaper option. The drawback is that it provides a square wave instead of a sine wave, but a square wave can still supply the correct RMS voltage. On-Off control was not chosen due to the fact that it not good for higher power applications.

USB Characteristics

- Easy to use.
- Built in functionality for all listed microcontroller options.
- Quick response time.

Serial Characteristics

- Slow data transfer.
- Transfers one bit at a time.
- Difficult to implement.
- Dependent on OS drivers. (Not easily compatible with different systems.)

Data Transfer Selection and Reasoning

Due to the faster transfer rate, ease of use, and built in functionality of USB, USB will be our preferred method of microcontroller to computer communication.

Display Options

Display Type	Price	Premade Library	Display Size
OLED	$$	No (some exceptions)	0.96 Inch
LCD	$	Yes	60×99x22 mm
Touch Screen	$$$	Yes	2.8 Inches

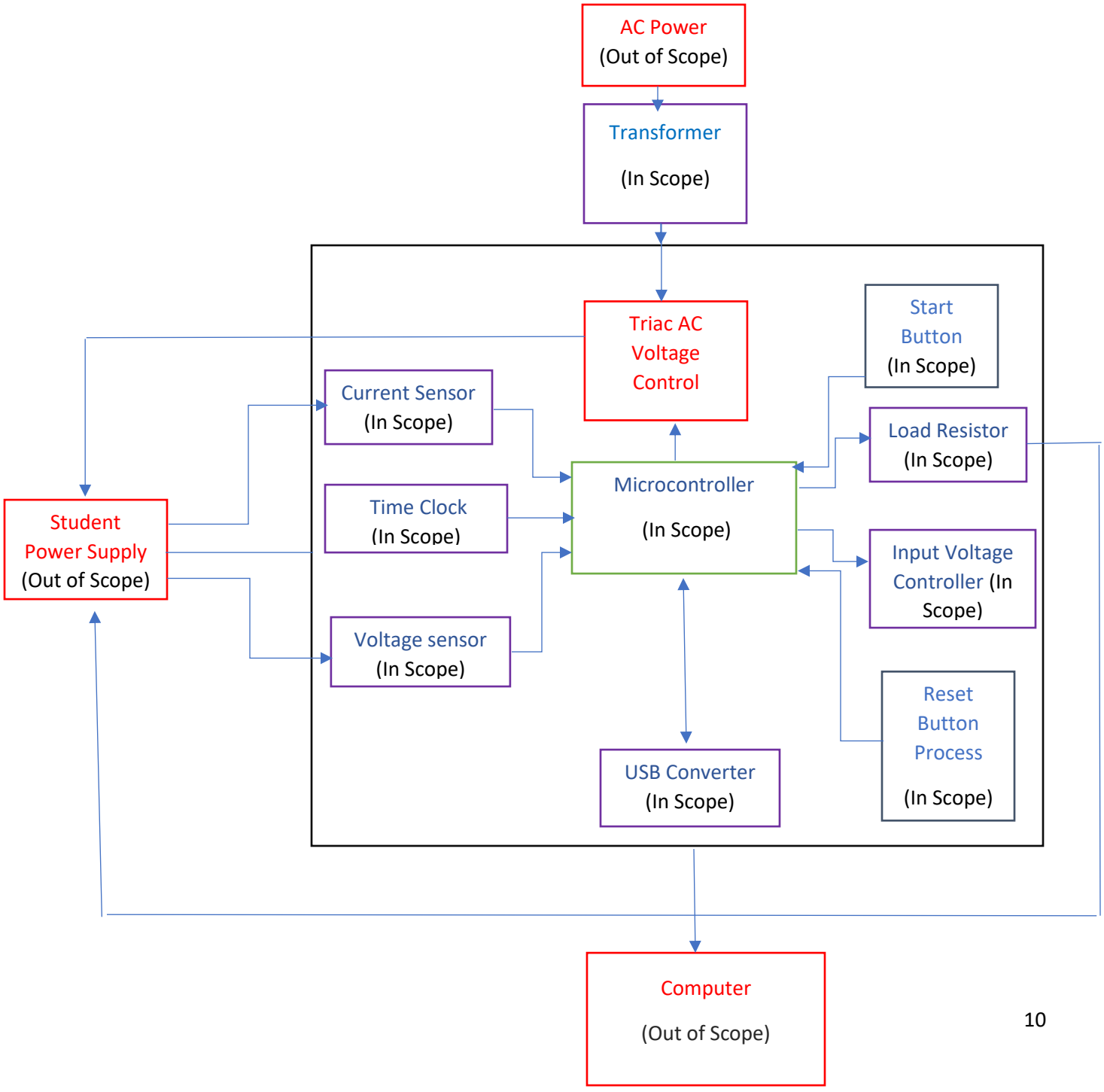
Display Selection and Reasoning

A LCD display will be used due to many factors. The LCD has the lowest cost of the screen options. The LCD also comes in more size options than the touch screen or OLED options. The OLED options typically come in a 0.96" size, with few exceptions. The LCD will also allow for future modifications better than the other options due to its ease of use.

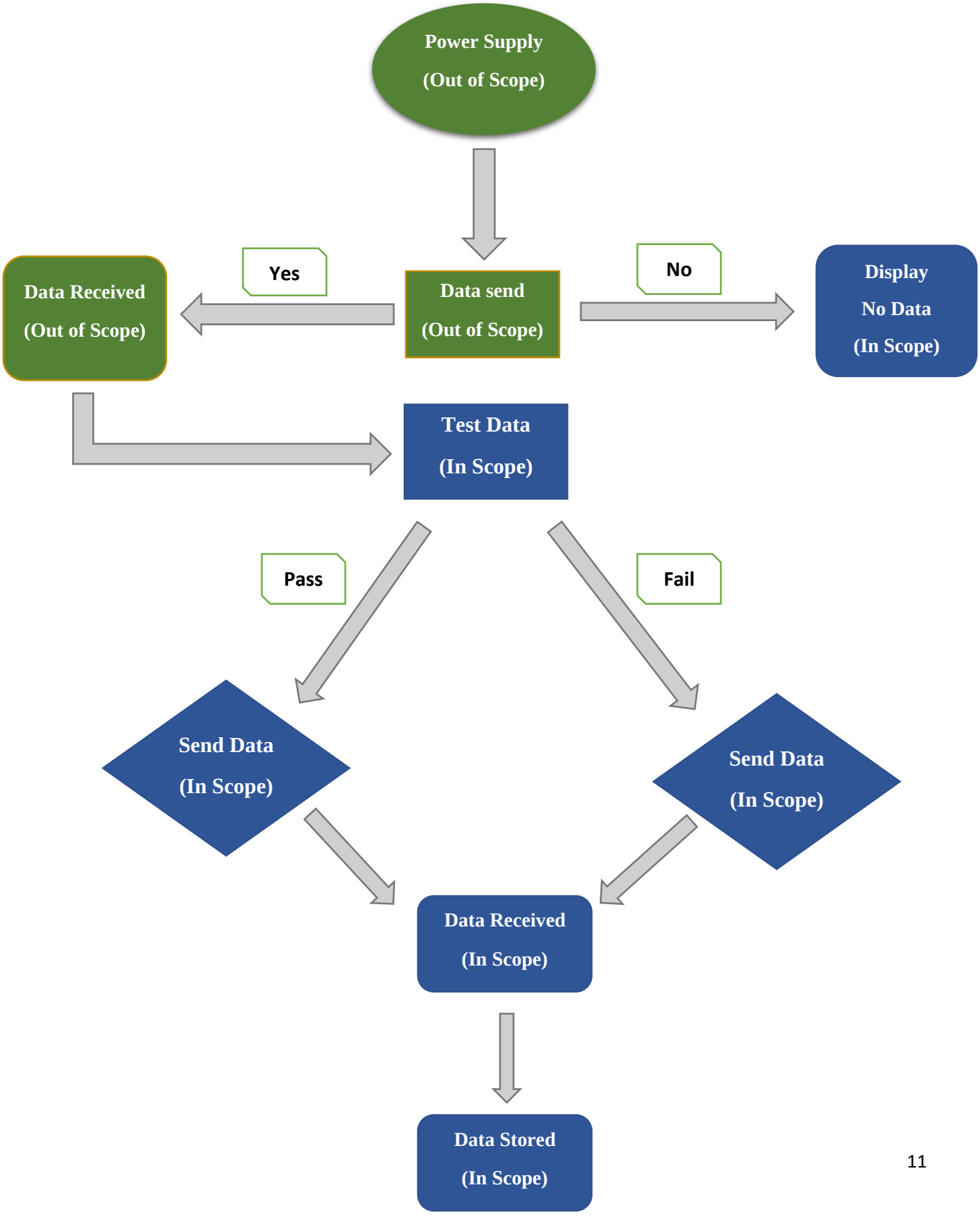
Cooling Selection and Reasoning

A fan will be used due to the heat produced in a constricted space. The power supply test unit must dissipate 15 watts. This heat would most likely be too much for passive cooling to handle. We will confirm or reject this decision depending on the thermo analysis results.

Hardware Block Diagram



Software flowchart overview.



Interface Design

The user interface will consist of a 20x4 LCD display with a 4x4 matrix keypad This setup will allow for user input of a wide variety, and a large display to show results and the current test being run. The LCD screen interface and keypad are shown below in figure 1. The LCD will instruct the user step by step through the test. The test is completely automated with the exception that the user must turn their potentiometer to adjust the voltage. This will also be instructed through the LCD.

Figure 1

Detailed Hardware Design – DC Measurement Circuit

There are four main circuits used in this project. They consist of a DC measurement, AC measurement, Triac, and zero crossing detection circuit. The DC measurement circuit is designed to take up to a 50V DC input and reduce it down to a 3V input. This is done to ensure that the Arduino has over voltage protection. The resistors were chosen by using the formula Vout = (Vin*R2)/(R1+R2). Vin was set to 50V and the desired output was 3V. There were many resistor options to give this output, but a 51.1k in series and a 10k and 5k in parallel were chosen due to the availability and pricing of these resistors. The circuit uses a relay to switch between the measurement circuit with the load and without the load. This allows for the power supply test unit to determine the voltage difference when a load is applied. Calibration of this circuit is done in the code when needed.

Ripple Measurement Circuit

The ripple measurement circuit is designed to cut out the DC signal and to amplify the ripple signal if it is below a certain threshold. This was accomplished by using a high pass RC filter in conjunction with an op amp. The RC filter eliminates the DC, while the op amp amplifies the signal for better resolution. Figure 3 shows the ripple voltage measurement circuit.

Figure 3

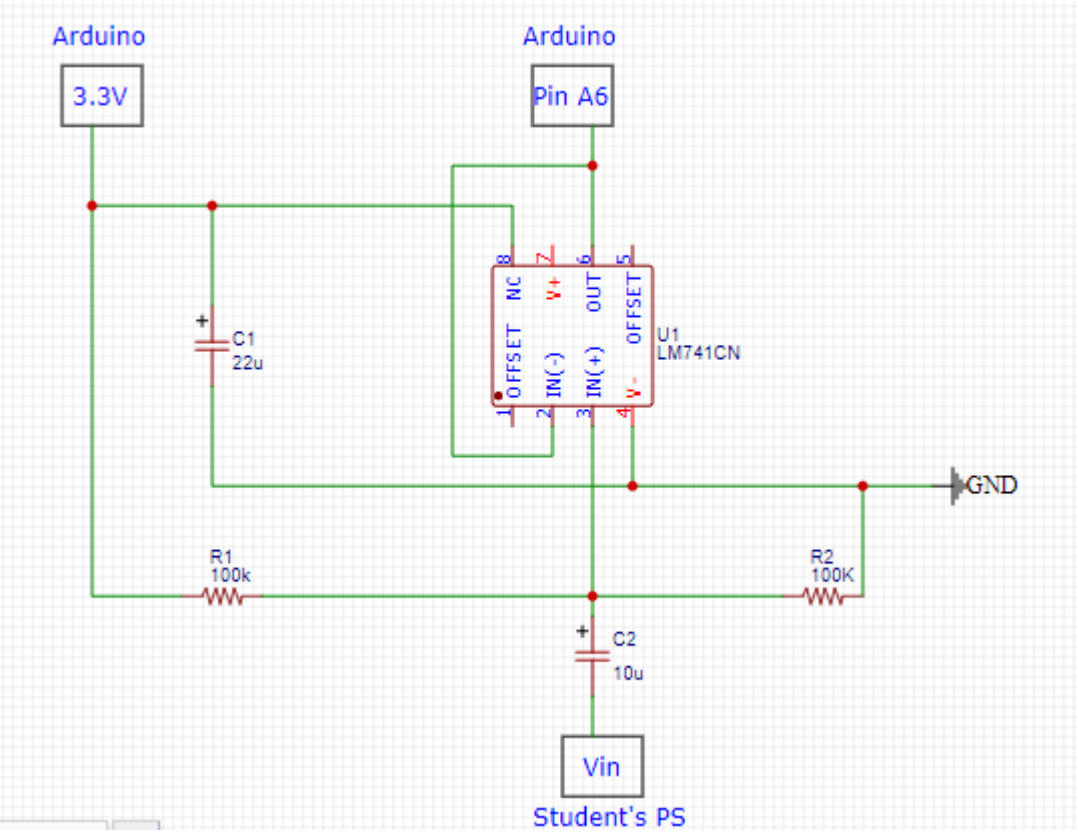


Figure 4

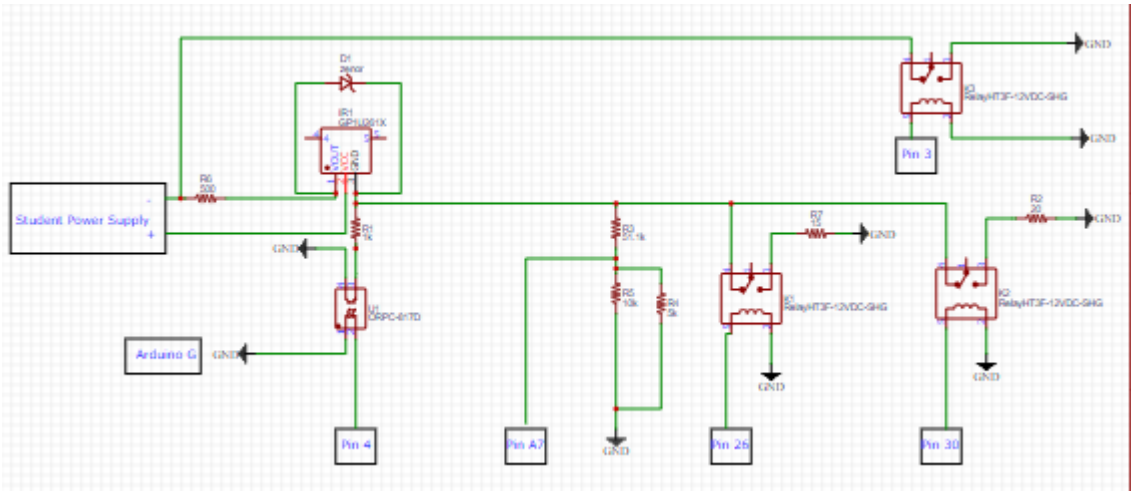


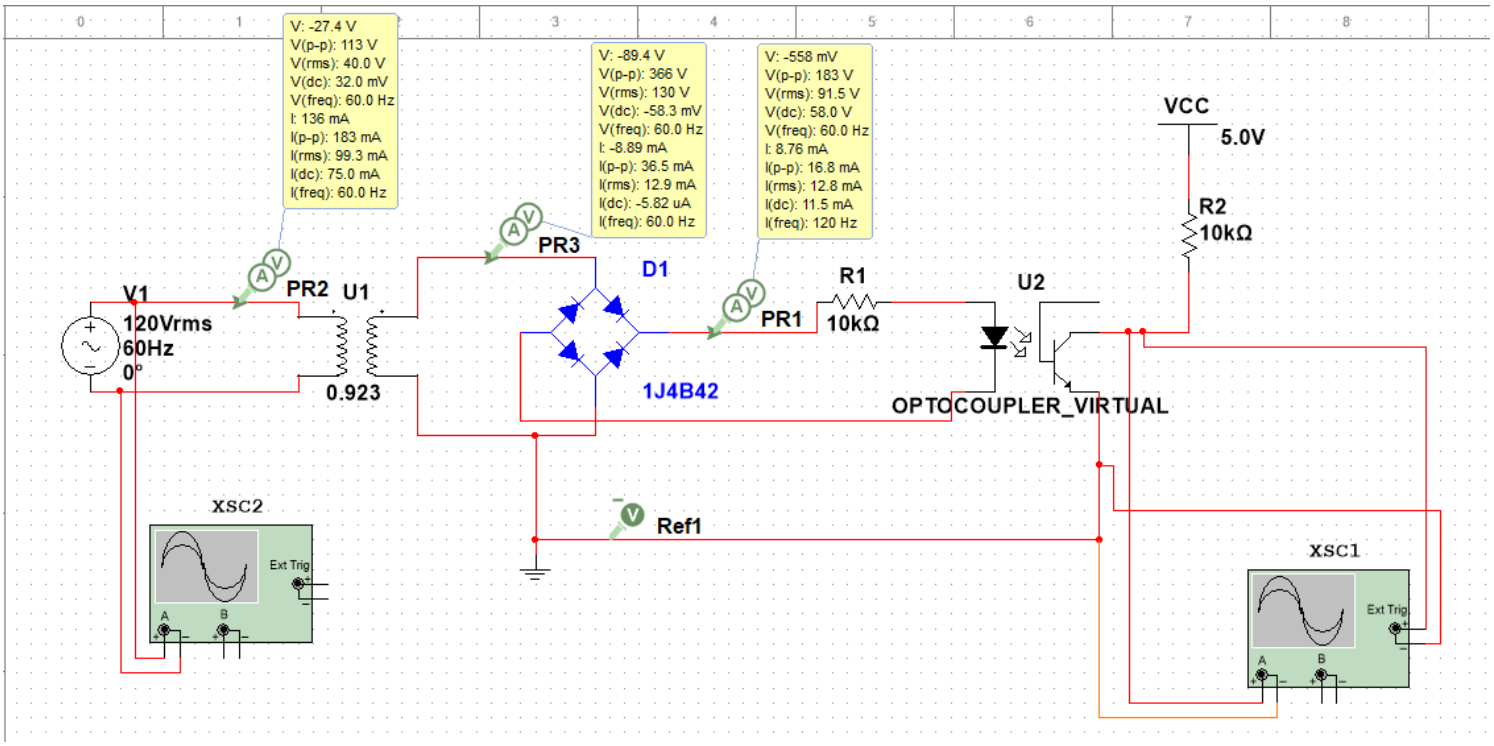
Figure 4 shows the connections that will be made to the student's power supply and to the Arduino. Relays are used to control when the grounds are tied together. Relays are also used to connect the 15 and 20 ohm resistors when needed. The DC measurement circuit is also included in figure, along with the reverse polarity protection.

Zero Crossing Detection Circuit

The zero-crossing detection circuit is used to determine when the AC sine wave crosses the zero-voltage point. The voltage source is sent through a full bridge rectifier to convert the signal to DC. The DC pulses are then sent to the opto-isolator to send out a 3.3V signal to the Arduino. The Arduino then uses this signal to determine the phase angle that the voltage source should be shut off at. The zero-crossing detection circuit is shown below in figure 6.

AC Voltage Control Circuit

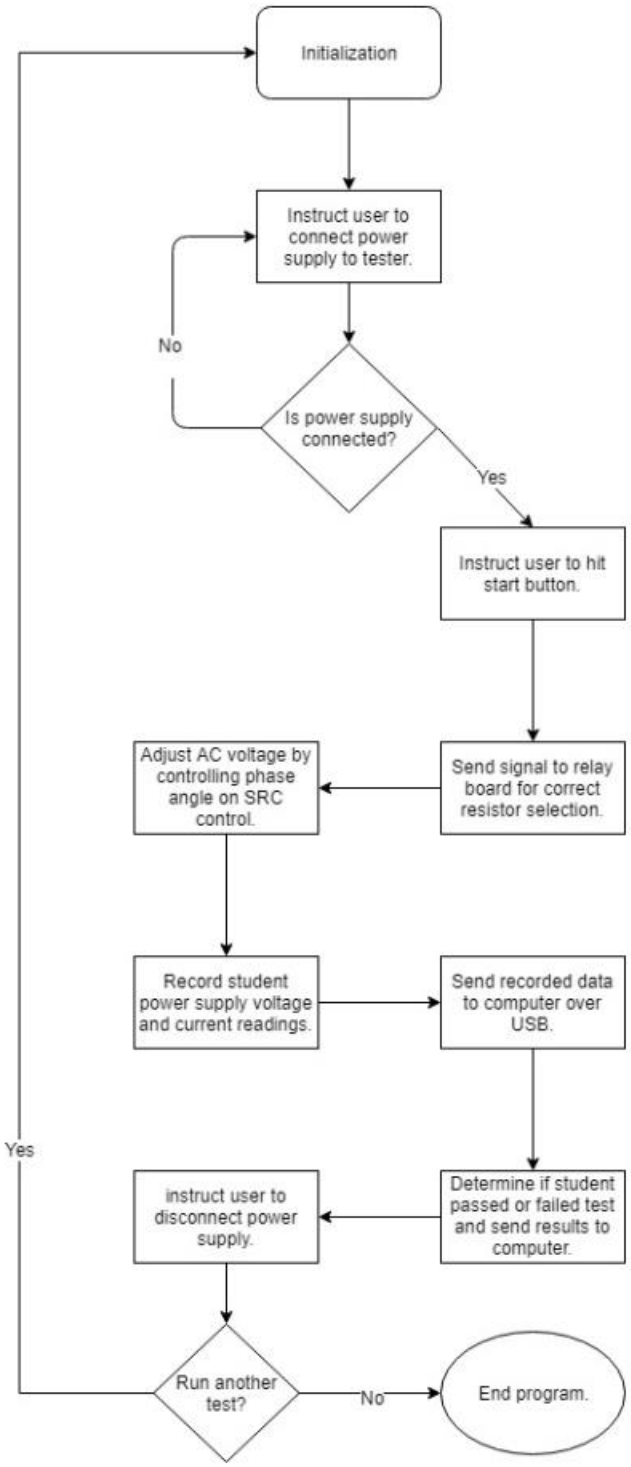
The Triac control circuit is also seen in figure 7. The Arduino sends out a PWM signal to the opto-isolator. The opto-isolator then transfers the signal over to the triac gate pin. This allows the Arduino to control the triac at high AC voltages without being damaged. The PWM and firing angle is calculated based on the zero voltage crossings found using the zero-crossing detection circuit.



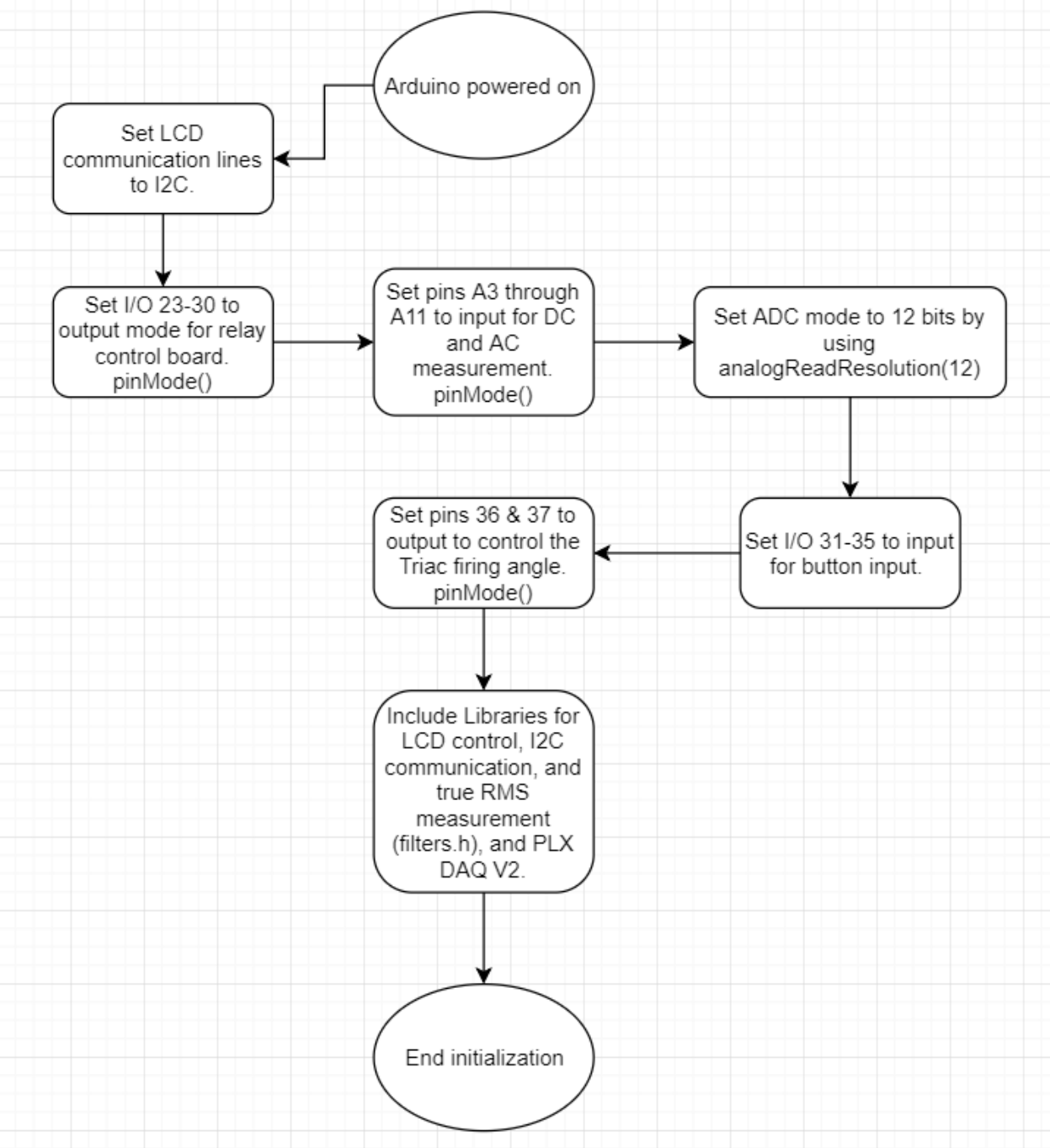
Input / Output Table

	A	B	C	D	E	F
1	**Pin #**	**Pinmode**	**Due/Nano**	**Analog/Digital**	**Corresponding Hardware**	**Corresponding Variable Name**
2	A0	Input	Due	Analog	ZMPT101B	ZMPT101B
3	A6	Input	Due	Analog	Ripple Voltage Circuit	Un-named
4	A7	Input	Due	Analog	DC Voltage Measurement Circuit	Un-named
5	4	Input	Due	Digital	Positive Polarity Optoisolator	Un-named
6	36	Input	Due	Digital	Keypad	rowPins
7	38	Input	Due	Digital	Keypad	rowPins
8	40	Input	Due	Digital	Keypad	rowPins
9	42	Input	Due	Digital	Keypad	rowPins
10	44	Input	Due	Digital	Keypad	colPins
11	46	Input	Due	Digital	Keypad	colPins
12	48	Input	Due	Digital	Keypad	colPins
13	50	Input	Due	Digital	Keypad	colPins
14	3	Output	Due	Digital	Relay	PositivePolarity
15	5	Output	Due	Digital	Nano	LowVoltage
16	6	Output	Due	Digital	Nano	LineVoltage
17	7	Output	Due	Digital	Nano	HighVoltage
18	8	Output	Due	Digital	Nano	minus
19	9	Output	Due	Digital	Nano	plus
20	10	Output	Due	Digital	Nano	AcOn
21	14	Output	Due	Digital	Nano	NanoReset
22	26	Output	Due	Digital	Relay	LowOhm
23	30	Output	Due	Digital	Relay	HighOhm
24	4	Input	Nano	Digital	Due	LowVoltage
25	5	Input	Nano	Digital	Due	LineVoltage
26	6	Input	Nano	Digital	Due	HighVoltage
27	7	Input	Nano	Digital	Due	minus
28	8	Input	Nano	Digital	Due	plus
29	9	Input	Nano	Digital	Due	AcOn
30	3	Input (interrupt)	Nano	Digital	zero crossing optoisolator	Un-named
31	11	Output	Nano	Digital	Triac	AC_pin

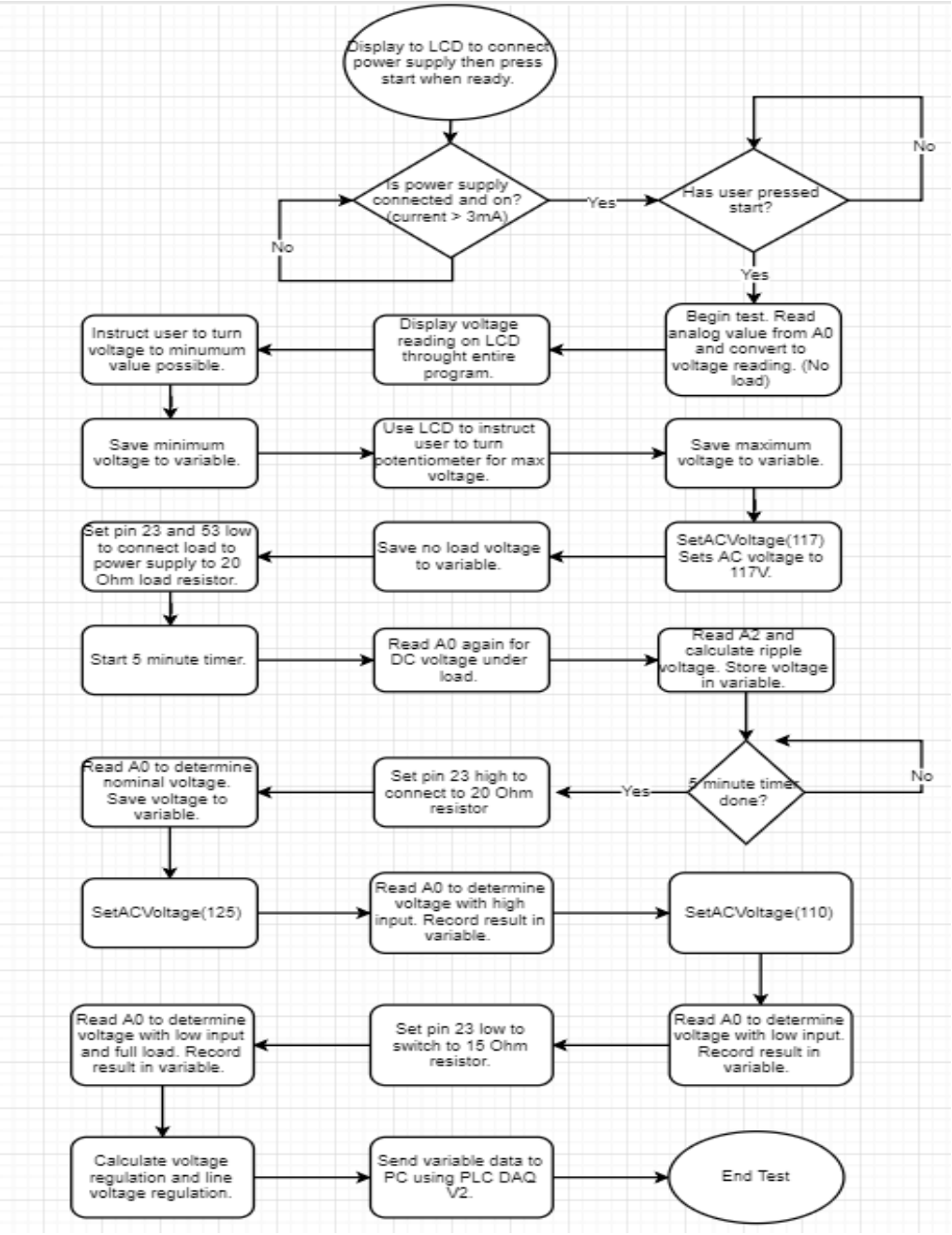
Software Flowchart – High Level

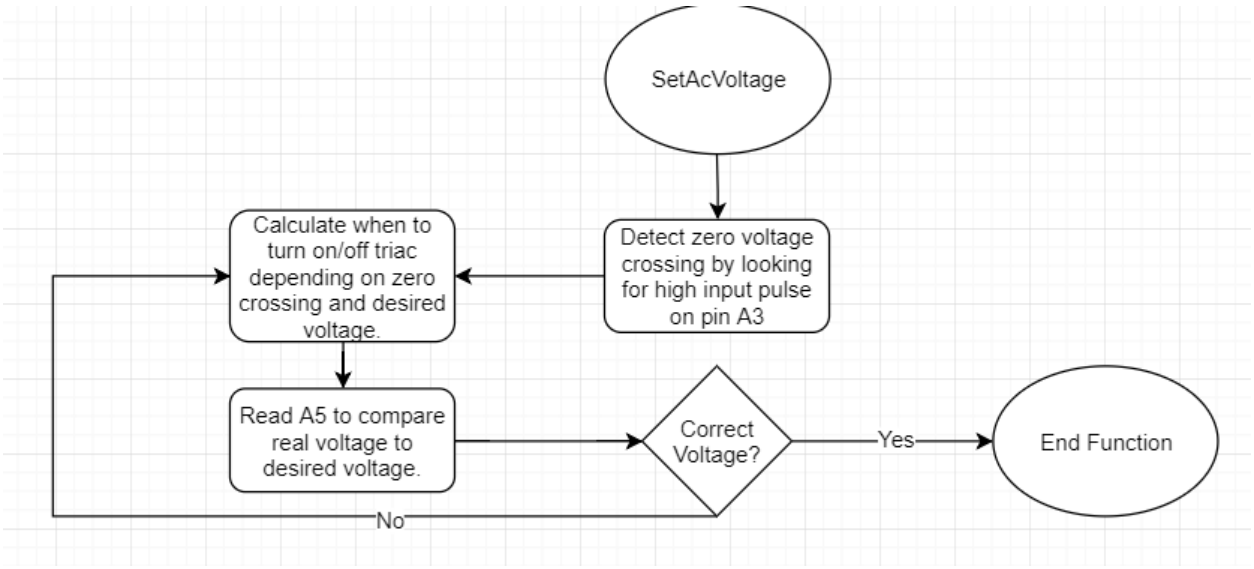


Initialization Flowchart



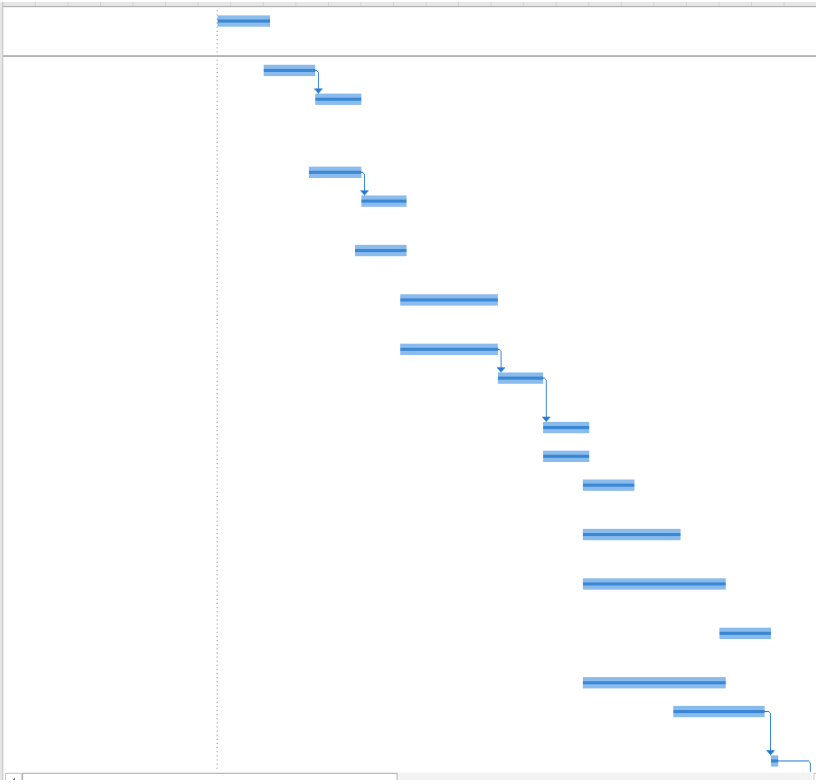
Run Test Flowchart





Gantt Chart

	Signed Group Guidelines	6 days	Tue 9/17/19	Tue 9/24/19		100%
	VOC and Scope	6 days	Tue 9/24/19	Tue 10/1/19		100%
	Signed Project Proposal and Charter	5 days	Wed 10/2/19	Tue 10/8/19	2	100%
	Draft Specifications	6 days	Tue 10/1/19	Tue 10/8/19		100%
	Signed Functional Specifications	5 days	Wed 10/9/19	Tue 10/15/19	4	100%
	Initial Project Timeline	6 days	Tue 10/8/19	Tue 10/15/19		100%
	Design Tradeoff Analysis	11 days	Tue 10/15/19	Tue 10/29/19		100%
	Block Diagram	11 days	Tue 10/15/19	Tue 10/29/19		100%
	High Level Design Draft	5 days	Wed 10/30/19	Tue 11/5/19	8	100%
	Test Plan Draft	5 days	Wed 11/6/19	Tue 11/12/19	9	100%
	High Level Design	5 days	Wed 11/6/19	Tue 11/12/19		100%
	DC Voltage Measurement	6 days	Tue 11/12/19	Tue 11/19/19		100%
	AC Voltage Control Circuit	11 days	Tue 11/12/19	Tue 11/26/19		100%
	AC Measurement Circuit	16 days	Tue 11/12/19	Tue 12/3/19		100%
	Reverse Polarity Protection Circuit	6 days	Tue 12/3/19	Tue 12/10/19		100%
	Low Level Design	16 days	Tue 11/12/19	Tue 12/3/19		100%
	Project Presentation File	10 days?	Tue 11/26/19	Mon 12/9/19		100%
	Presentation	1 day	Wed 12/11/19	Wed 12/11/19	17	100%



Bill of Materials

1+A3:M24A3:M26A3:M25A3:M26A3:M24	EL-SM-005	2	ELEGOO 8 Channel DC 5V Relay Module with Optocoupler for Arduino UNO	oupler-Arduino-Raspberry/dp/B01HCFJC0Y/ref=sr_1_5?keywords	$8.99
2	g18122000ux0031	2	uxcell 2 Pcs Aluminum Case Resistor 100W 15 Ohm Wirewound Green for LED Replacement Converter 100W 15RJ	placement-Converter/dp/B07FS4YBTX/ref=sr_1_20?keywords=1	$10.15
3	RNMF14FTC30K0	1	30 kOhms ±1% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Metal Film	n/product-detail/en/RNMF14FTC30K0/S30KCACT-ND/261748	$0.10
4	RNV14FAL470K	1	470 kOhms ±1% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, High Voltage, Moisture Resistant, Pulse Withstanding, Safety Metal Film	oduct-detail/en/stackpole-electronics-inc/RNV14FAL470K/RNV14	$0.26
5	FA16C0G2A473JRU06	1	Multilayer Ceramic Capacitors MLCC - Leaded 100V 0.047uF 5% AEC-Q200 Radial C0G	il/TDK/FA16C0G2A473JRU06?qs=sGAEpiMZZMuMW9TJLBQkXh	$0.80
6	3549S-1AA-203A	2	20k Ohm 1 Gang Linear Panel Mount Potentiometer None 10 Turn Wirewound 2W Solder Lug	.com/product-detail/en/bourns-inc/3549S-1AA-203A/3549S-1AA-	$18.51
7	MBB02070C5112FCT00	2	51.1 kOhms ±1% 0.6W Through Hole Resistor Axial Automotive AEC-Q200 Metal Film	product-detail/en/vishay-bc-components/MBB02070C5112FCT00/	$0.29
8	G5V-2-DC5	1	General Purpose Relay DPDT (2 Form C) Through Hole 2A 5VDC	y.com/product-detail/en/omron-electronics-inc-emc-div/G5V-2-DC	$2.88
9	20x4 SerLCD - Black on RGB 3.3V	1	SunFounder IIC I2C TWI Serial 2004 20x4 LCD Module Shield for Arduino	https://www.sparkfun.com/products/14074	$24.95
10	DEV-11589	1	Microcontroller: AT91SAM3X8E Operating Voltage: 3.3V Recommended Input Voltage: 7-12V Min-Max Input Voltage: 6-20V Digital I/O Pins: 54 (of which 12 provide PWM output) Analog Input Pins: 12 Analog Outputs Pins: 2 Total DC Output Current on all I/O lines: 130 mA DC Current for 3.3V Pin: 800 mA DC Current for 5V Pin: 800 mA Flash Memory: 512 KB all available for the user applications SRAM: 96 KB (two banks: 64KB and 32KB)Clock Speed: 84 MHz	https://www.sparkfun.com/products/11589	$41.95
11	LM741CNNS/NOPB-ND	1	IC OPAMP GP 1 CIRCUIT 8DIP (General Purpose Amplifier 1 Circuit 8-PDIP)	n/product-detail/en/texas-instruments/LM741CN-NOPB/LM741CI	$0.87
12	1N4732AFSTR-ND	1	Zener Diode 4.7V 1W ±5% Through Hole DO-41	y.com/product-detail/en/on-semiconductor/1N4732ATR/1N4732AI	$0.26
13	160-2045-2-ND	1	Optoisolator Triac Output 5000Vrms 1 Channel 6-SMD	ikey.com/product-detail/en/lite-on-inc/MOC3021S-TA1/160-2045	$0.26
14	1740-1020-ND	1	TRIAC Logic - Sensitive Gate 600V 4A Through Hole TO-220AB	n/product-detail/en/ween-semiconductors/BT136-600E-L01127/17	$0.60
15	4N25VS-ND	1	Optoisolator Transistor with Base Output 5000Vrms 1 Channel 6-DIP	com/product-detail/en/vishay-semiconductor-opto-division/4N25/4N	$0.26
16	ZMPT101B	1	2mA Active Single Phase Voltage Transformer Module AC Output Voltage Sensor for Arduino	n=google&aff_short_key=UneMJZVf&&albagn=888888&albcp=158	$0.89
17	ACS712	1	Hall Current Sensor Module ACS712 module 5A Hall Current Sensor Module for arduino 5A/20A/30A ACS712	google&aff_short_key=UneMJZVf&&albagn=888888&albcp=1582	$1.01
18	CFR-25JB-52-1K	2	1 kOhms ±5% 0.25W, 1/4W Through Hole Resistor	ey.com/product-detail/en/CFR-25JB-52-1K/1.0KQBK-ND/96/?item	$0.20
19	CF14JT3K30	1	3.3 kOhms ±5% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film	om/product-detail/en/CF14JT3K30/CF14JT3K30CT-ND/1830362/	$0.10
20	CF14JT10K0CT-ND	2	10 kOhms ±5% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film	om/product-detail/en/CF14JT10K0/CF14JT10K0CT-ND/1830374/	$0.20
21	CF12JT470KCT-ND	2	470 kOhms ±5% 0.5W, 1/2W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film	om/product-detail/en/CF12JT470K/CF12JT470KCT-ND/1830538/	$0.20
22	CMF29.4KHFCT-ND	2	29.4 kOhms ±1% 0.5W, 1/2W Through Hole Resistor Axial Flame Retardant Coating, Moisture Resistant, Safety Meta	om/product-detail/en/vishay-dale/CMF5529K400FHEB/CMF29.4K	$1.16
23	CT3050-ND	2	1k Ohm 1 Gang Linear Panel Mount Potentiometer None 1 Turn Carbon 0.5W, 1/2W Solder Lug	om/product-detail/en/cts-electrocomponents/450T328S102A1A1/	$7.66
24	NTE5332	1	Bridge Rectifier - Full Wave Single Phase 600V 1amp 4-pin DIP Package	s/PRODUCT/NTE5332?gclid=EAIaIQobChMI3NHGyK6s5gIVUSCtB	$1.92
25	CMF5.0KAA-ND	1	5 kOhms ±5% 0.5W, 1/2W Through Hole Resistor Axial Flame Retardant Coating, Moisture Resistant, Safety Metal Fil	y.com/product-detail/en/vishay-dale/CMF075K0000JNEK/CMF5.0	$0.42
26	PPC51KW-2CT-ND	1	51 kOhms ±5% 2W Through Hole Resistor Axial Automotive AEC-Q200, Flame Retardant Coating, Safety Metal Film	roduct-detail/en/vishay-bc-components/PR02000205102JR500/PF	$0.45
27	CF14JT100KCT-ND	1	100 kOhms ±5% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film	n/product-detail/en/stackpole-electronics-inc/CF14JT100K/CF14J	$0.10
28	HS150 20R F	1	Wirewound Resistors - Chassis Mount 150W 20 ohm1%	etail/ARCOL-Ohmite/HS150-20R-F?qs=sGAEpiMZZMtbXrlkmrvidL	$13.31
29	490-14507-ND	1	22µF ±10% 25V Ceramic Capacitor X7S Radial	n/product-detail/en/murata-electronics/RDEC71E226K3K1H03B/4	$1.82
30	CF14JT150RCT-ND	1	150 Ohms ±5% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film	n/product-detail/en/stackpole-electronics-inc/CF14JT150R/CF14J	$0.10
31	739W-X2/03	1	Female 125VAC receptacle	www.digikey.com/product-detail/en/qualtek/739W-X2-03/Q228-ND/	$1.05

Code for Due

```
#include <SerLCD.h>
#include <Filters.h>                                //Library to use
#include <Wire.h>
#include <Keypad.h>

#define ZMPT101B A0                                 //Analog input

const int numReadings = 10;
const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

char studentIdinput[4] = {0};
char key[4];
char Main[4] = {'4', '6', '2', '7'}; //array to hold keypad password
byte currentLength = 0;
char inputArray[4]; //array to gather user keypad presses
byte rowPins[ROWS] = {42, 40, 38, 36};
byte colPins[COLS] = {50, 48, 46, 44};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

float readings[numReadings];      // the readings from the analog input
int readIndex = 0;               // the index of the current reading
long total = 0;                  // the running total
float average = 0;
```

```
float testFrequency = 120;                     // test signal frequency (Hz)
float windowLength = 40 / testFrequency;     // how long to average the signal, for statistist, changing this can have drastic effect
// Test as you need

long RawValue = 0;
float Volts_TRMS;     // estimated actual voltage in Volts

float intercept = 0;  // to be adjusted based on calibration testin
float slope = 0.55; // measure with multimeter and compare to non calibrated result. Use multiplication or division to get this value.
long milli_time = millis();
long milli_time1 = millis();
float DcVoltage;
volatile long value;
int password = 0;
int code = 1570;  //Password
int tot, i1, i2, i3, i4;
char c1, c2, c3, c4;
byte StudentIdentified = 0;
SerLCD lcd; // Initialize the library with default I2C address 0x72
/* How to get the intercept and slope. First keep them like above, intercept=0 and slope=1,
   also below keep displaying Calibrated and non calibrated values to help you in this process.
   Put the AC input as 0 Volts, upload the code and check the serial monitor, normally you should have 0
   if you see another value and it is stable then the intercept will be the opposite of that value
   Example you upload first time and then you see a stable 1.65V so the intercept will be -1.65
   To set the slope now you need to put the voltage at something higher than 0, and measure that using your reference TRMS multimeter
   upload the new code with the new intercept and check the value displayed as calibrated values
   Slope = (Measured values by multimeter)/(Measured values by the code)
   Place your new slope and reupload the code, if you have problems with calibration try to adjust them both
   or add a new line to calibrate further
   Slope and intercept have nothing to do with the TRMS calculation, it's just an adjustement of the line of solutions
*/

unsigned long printPeriod = 1000; //Measuring frequency, every 1s
unsigned long previousMillis = 0;
unsigned long printPeriod1 = 10000; //Measuring frequency, every 5s, can be changed // change back to 20000
unsigned long previousMillis1 = 0;
byte polarity = 0;
float DcMin = 0;
float DcMax = 0;
byte DcDone = 0;
float DcScaleHigh = 16.7700;
float DcScaleLow = 22.8700;
float DcScale = 0;
byte DcTest = 0;
int PositivePolarity = 3; //Digital Output to Relay connecting grounds when there is positive polarity
int DcFlag = 0;
int period = 8000;
unsigned long time_now = 0;
long rippleValue = 0;
float rippleVoltage = 0;
float maxRipple;
float minRipple;
float slope1;
float intercept1;
long DcTestDuration = 15000;
int minus = 8;
int AcOn = 10;
int plus = 9;
int HighVoltage = 7;
int LineVoltage = 6;
int LowVoltage = 5;
float NominalVoltageSpec = 0;
int LowOhm = 26;
int HighOhm = 30;
```

```
float VoutLow = 0;
float VoutHigh = 0;
byte LineFlag = 0;
byte LoadRegFlag = 0;
float LowVoltageSpec = 0;
long FullLoadDuration = 25000; // change back to 260000 for long test
long DcTestDuration1 = 15000; // change back to 40000 for long test
float VoutFullLoad1 = 0;
float VoutFullLoad2 = 0;
int NanoReset = 14; // Digital output to reset nano when due is reset
float VoltageRegulation = 0;
float VLR = 0;
int CalcFlag = 0;
int ExcelFlag = 0;

RunningStatistics inputStats; //This class collects the value so we can apply some functions

float mapF(float x, float in_min, float in_max, float out_min, float out_max) {
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
float ConvertVoltage(float value = 0) {
  return mapF(value, 51.0, 977.0, 1.21, 12.87);
}
float ConvertVoltage(long value = 0) {
  long valuel = value;
  return mapF(valuel, 51.0, 977.0, 1.21, 12.87);
}


void setup() {
  Serial.begin(115200);    // start the serial port
  Serial.println("Serial started");
```

```
130  inputStats.setWindowSecs( windowLength );
131  // initialize all the readings to 0:
132  analogReadResolution(12);
133  //for (int thisReading = 0; thisReading < numReadings; thisReading++) {
134  //readings[thisReading] = 0;
135  Serial.println("CLEARDATA");        //This string is defined as a
136  // to clear all the rows and columns
137  Serial.println("LABEL,Computer Time,Student ID,Minimum DC Output,Maximum DC Output,Line Regulat
138  pinMode(4, INPUT_PULLUP);
139  pinMode(3, OUTPUT);
140  Wire.begin();
141  lcd.begin(Wire);  //By default .begin() will set I2C SCL to Standard Speed mode of 100kHz
142  Wire.setClock(400000); //Optional - set I2C SCL to High Speed Mode of 400kHz
143  lcd.setBacklight(0xA020F0); //violet
144  lcd.setCursor(0, 1);
145  pinMode(minus, OUTPUT);
146  pinMode(AcOn, OUTPUT);
147  pinMode(plus, OUTPUT);
148  pinMode(HighVoltage, OUTPUT);
149  pinMode(LineVoltage, OUTPUT);
150  pinMode(LowVoltage, OUTPUT);
151  digitalWrite(minus, LOW);
152  digitalWrite(AcOn, LOW);
153  digitalWrite(plus, LOW);
154  digitalWrite(HighVoltage, LOW);
155  digitalWrite(LineVoltage, LOW);
156  digitalWrite(LowVoltage, LOW);
157  pinMode(LowOhm, OUTPUT);
158  pinMode(HighOhm, OUTPUT);
159  digitalWrite(LowOhm, LOW);
160  digitalWrite(HighOhm, LOW);
161  pinMode(NanoReset, OUTPUT);
```

```
  digitalWrite(NanoReset, HIGH);
  delay(1500);
  digitalWrite(NanoReset, LOW);
  delay(1500);
  digitalWrite(NanoReset, HIGH);

  //}
}

void loop() {
  digitalWrite(3, LOW);
  lcd.clear();
  digitalWrite(LowOhm, LOW);
  digitalWrite(HighOhm, LOW);
  digitalWrite(AcOn, LOW);
  digitalWrite(plus, LOW);
  digitalWrite(HighVoltage, LOW);
  digitalWrite(LineVoltage, LOW);
  digitalWrite(LowVoltage, LOW);
  if (password == 0)
  {
    CheckPassword();
  }

  if (StudentIdentified == 0)
  {
    GetStudentId();
  }

  if (DcTest == 1)
  {
    DcMinMax();
```

```
193     DcMinMax();
194   }
195   if (LineFlag == 1)
196   {
197     LineRegulation();
198   }
199
200   if (LoadRegFlag == 1)
201   {
202     LoadRegulation();
203   }
204
205   if (CalcFlag == 1)
206   {
207     Calculations();
208   }
209
210   if (ExcelFlag == 1)
211   {
212     SendToExcel();
213   }
214
215   lcd.clear();
216   lcd.setCursor(0, 2);
217   lcd.print("Press any key to    ");
218   lcd.print("test again");
219   char  keypressed = keypad.waitForKey();
220
221   if (keypressed != NO_KEY) {
222     // do nothing
223   }
224
```

```
225   password = 0;
226   StudentIdentified = 0;
227
228 }
229
230 void  ReadVoltage() {
231   RawValue = analogRead(ZMPT101B);  // read the analog in value:
232   inputStats.input(RawValue);       // log to Stats function
233
234   if ((unsigned long)(millis() - previousMillis) >= printPeriod)
235     previousMillis = millis();   // update time
236
237     Volts_TRMS = inputStats.sigma() * slope + intercept;
238     //      Volts_TRMS = Volts_TRMS*0.979;              //Further
239
240     Serial.print("Non Calibrated: ");
241     Serial.print("\t");
242     Serial.print(inputStats.sigma());
243     Serial.print("\t");
244     Serial.print("Calibrated: ");
245     Serial.print("\t");
246     Serial.println(Volts_TRMS);
247     Serial.print("\t");
248     Serial.print(average);
249
250   }
251 }
252
253 void DcMinMax() {
254
255   digitalWrite(AcOn, HIGH);
256   digitalWrite(LineVoltage, HIGH);
```

```
255   digitalWrite(AcOn, HIGH);
256   digitalWrite(LineVoltage, HIGH);
257   lcd.clear();
258   if (digitalRead(PositivePolarity == 1)) {
259     digitalWrite(3, HIGH);
260   }
261   while (PositivePolarity == 0) {
262     digitalRead(PositivePolarity);
263     lcd.setCursor(0, 2);
264     lcd.print("Raise Voltage to    ");
265     lcd.print("halfway point");
266   }
267   PolarityCheck();
268   lcd.setCursor(0, 0);
269   lcd.clear();
270   lcd.setCursor(0, 0);
271   lcd.print("Set power supply to ");
272   lcd.print("1.2 VDC  press any  key when ");
273   lcd.print("ready");
274
275   while ((DcVoltage < 0.2) || (DcVoltage > 1.4))
276   {
277
278     for (int i = 0; i < 50 ; i++) {
279       value += analogRead(A7);
280     }
281     value = value / 50;
282
283     DcVoltage = ConvertVoltage(value);
284     lcd.setCursor(0, 3);
285     lcd.print(DcVoltage);
286     lcd.print(" VDC");
```

```
287   }
288   for (int i = 0; i < 50 ; i++) {
289     value += analogRead(A7);
290   }
291   value = value / 50;
292
293   DcVoltage = ConvertVoltage(value);
294   lcd.setCursor(0, 3);
295   lcd.print(DcVoltage);
296   lcd.print(" VDC");
297   char  keypressed = keypad.waitForKey();  // here all pro
298
299   if (keypressed != NO_KEY) {
300     // do nothing
301   }
302   lcd.clear();
303   lcd.setCursor(0, 2);
304   lcd.print("Correct Voltage");
305   delay(3000);
306   lcd.clear();
307   long break_time = 0;
308   break_time = millis() + DcTestDuration;
309   while (millis() <= break_time) {
310
311     float average = 0;
312
313     DcVoltage = 0;
314     value = 0;
315
316     for (int i = 0; i < 50 ; i++) {
317       value += analogRead(A7);
318     }
```

```
    value = value / 50;

    DcVoltage = ConvertVoltage(value);


    if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
      previousMillis = millis();   // update time every second
      lcd.setCursor(0, 0);
      lcd.print("Measuring Min VDC");
      lcd.setCursor(0, 2);
      lcd.print(DcVoltage);
      Serial.println(value);
      Serial.println(DcVoltage);
      lcd.setCursor(5 , 2);
      lcd.print(" Volts DC      ");
    }
  }

  DcMin = DcVoltage;
  lcd.clear();
  keypressed = keypad.getKey();  //The getKey fucntion keeps the program runi
  lcd.print("Set power supply to ");
  lcd.print("15V and press any   ");
  lcd.print("key when ready");
  while ((DcVoltage > 12.1) || (DcVoltage < 11.93))
  {
    for (int i = 0; i < 50 ; i++) {
      value += analogRead(A7);
    }
    value = value / 50;

    DcVoltage = ConvertVoltage(value);
```

```
    DcVoltage = ConvertVoltage(value);
    lcd.setCursor(0, 3);
    lcd.print(DcVoltage);
    lcd.print(" VDC");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Voltage correct,    Press any");
  lcd.print("Key");
  keypressed = keypad.getKey();  //The getKey fucntion keeps th
  keypressed = keypad.waitForKey();  // here all programs are s

  if (keypressed != NO_KEY) {
    // do nothing
  }


  break_time = 0;
  PolarityCheck();
  while (PositivePolarity == LOW) {
    PolarityCheck();
  }
  lcd.clear();

  break_time = millis() + DcTestDuration;
  while (millis() <= break_time) {

    float average = 0;

    for (int i = 0; i < 50 ; i++) {
      value += analogRead(A7);
    }
```

```
383
384     DcVoltage = ConvertVoltage(value);
385
386
387     if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
388       previousMillis = millis();   // update time every second
389       lcd.setCursor(0, 2);
390       lcd.print(DcVoltage);
391       lcd.setCursor(5 , 2);
392       lcd.print(" Volts DC      ");
393     }
394     PolarityCheck();
395   }
396
397   DcMax = DcVoltage;
398   lcd.clear();
399   lcd.setCursor(0, 0);
400   lcd.print("Set power supply to ");
401   lcd.print("15V. Press any key  ");
402   lcd.print("when ready ");
403   while ((DcVoltage > 12.1) || (DcVoltage < 12))
404   {
405     for (int i = 0; i < 50 ; i++) {
406       value += analogRead(A7);
407     }
408     value = value / 50;
409
410     DcVoltage = ConvertVoltage(value);
411     lcd.setCursor(12, 3);
412     lcd.print(DcVoltage);
413     lcd.print(" VDC");
414   }
```

```
413     lcd.print(" VDC");
414   }
415   lcd.clear();
416   lcd.setCursor(0, 0);
417   lcd.print("Voltage correct,    Press any");
418   lcd.print("Key");
419   keypressed = keypad.waitForKey();  // here all programs are stoppe
420
421   if (keypressed != NO_KEY) {
422     // do nothing
423   }
424
425   digitalWrite(HighOhm, HIGH);
426   break_time = 0;
427   PolarityCheck();
428   while (PositivePolarity == LOW) {
429     PolarityCheck();
430   }
431   lcd.clear();
432
433   break_time = millis() + DcTestDuration;
434   while (millis() <= break_time) {
435
436     float average = 0;
437
438     for (int i = 0; i < 50 ; i++) {
439       value += analogRead(A7);
440     }
441     value = value / 50;
442
443     DcVoltage = ConvertVoltage(value);
```

```
447         previousMillis = millis();    // update time every second
448         lcd.setCursor(0, 2);
449         lcd.print(DcVoltage);
450         lcd.setCursor(5 , 2);
451         lcd.print(" Volts DC      ");
452       }
453       PolarityCheck();
454     }
455
456     NominalVoltageSpec = DcVoltage;
457     digitalWrite(HighOhm, LOW);
458     digitalWrite(LineVoltage, LOW);
459
460     DcTest = 0;
461     lcd.clear();
462     LineFlag = 1;
463
464
465   }
466
467
468   void CheckPassword()
469   {
580
581   void GetStudentId()
582   {
606
607   void PolarityCheck()
608   {
609     if (digitalRead(4) == LOW)
610     {
611       digitalWrite(3, HIGH);
```

```
    digitalWrite(3, HIGH);
    polarity = 1;
  }
  else
  {
    digitalWrite(3, LOW);
    polarity = 0;
  }
  int z = 0;
  while (digitalRead(4) == HIGH) {

    for (z; z <= 0; z++) {
      lcd.clear();
    }
    lcd.setCursor(0, 2);
    lcd.print("Check polarity");

    if (digitalRead(4) == LOW)
    {
      digitalWrite(3, HIGH);
      polarity = 1;
      delay(10);
    }
    else
    {
      digitalWrite(3, LOW);
      polarity = 0;
    }
  }

}

void DcRippleMeasurement()
```

```
        --
{
  rippleValue = analogRead(A6);  // read the analog in value:
  inputStats.input(rippleValue);       // log to Stats function

  if ((unsigned long)(millis() - previousMillis) >= printPeriod) { //We calculate and display ev
    previousMillis = millis();   // update time

    rippleVoltage = inputStats.sigma() * slope + intercept;
    lcd.setCursor(0, 2);
    lcd.print(rippleVoltage);

  }
}

void LineRegulation()
{
  delay(50);
  int break_time = 0;
  digitalWrite(HighVoltage, HIGH);
  lcd.clear();
  lcd.setCursor(0, 2);
  lcd.print("Line Regulation Test");
  while ((DcVoltage > 12.2) || (DcVoltage < 11.9)) {
    for (int i = 0; i < 50 ; i++) {
      value += analogRead(A7);
    }
    value = value / 50;

    DcVoltage = ConvertVoltage(value);
    lcd.setCursor(0, 3);
    lcd.print("Set to 15 ");
    lcd.setCursor(11, 3);
```

```
676     lcd.print(DcVoltage);
677     lcd.print(" V");
678   }
679
680   lcd.clear();
681   lcd.setCursor(0, 2);
682   lcd.print("Line Regulation Test");
683
684   digitalWrite(HighOhm, HIGH);
685   break_time = 0;
686   PolarityCheck();
687
688   break_time = millis() + DcTestDuration;
689   while (millis() <= break_time) {
690
691     float average = 0;
692
693
694     for (int i = 0; i < 50 ; i++) {
695       value += analogRead(A7);
696     }
697     value = value / 50;
698
699     DcVoltage = ConvertVoltage(value);
700
701
702     if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
703       previousMillis = millis();   // update time every second
704       lcd.setCursor(0, 3);
705       lcd.print(DcVoltage);
706       lcd.setCursor(5 , 2);
707       lcd.print(" Volts DC      ");
708     }
```

```
709       PolarityCheck();
710     }
711     VoutHigh = DcVoltage;
712
713
714     digitalWrite(HighVoltage, LOW);
715     delay(100);
716     digitalWrite(LowVoltage, HIGH);
717
718
719     lcd.clear();
720     lcd.setCursor(0, 2);
721     lcd.print("Line Regulation Test");
722
723     break_time = 0;
724     PolarityCheck();
725
726     break_time = millis() + DcTestDuration;
727     while (millis() <= break_time) {
728
729       float average = 0;
730
731
732       for (int i = 0; i < 50 ; i++) {
733         value += analogRead(A7);
734       }
735       value = value / 50;
736
737       DcVoltage = ConvertVoltage(value);
738
739
740       if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
```

```
742       lcd.setCursor(0, 3);
743       lcd.print(DcVoltage);
744       lcd.setCursor(5 , 3);
745       lcd.print(" Volts DC      ");
746     }
747     PolarityCheck();
748   }
749   VoutLow = DcVoltage;
750   digitalWrite(HighOhm, LOW);
751   LineFlag = 0;
752   LoadRegFlag = 1;
753 }
754
755 void LoadRegulation()
756 {
757   int break_time = 0;
758   value = 0;
759   PolarityCheck();
760   lcd.clear();
761   lcd.setCursor(0, 2);
762   lcd.print("Load Regulation Test");
763   digitalWrite(LowOhm, HIGH);
764
765   break_time = 0;
766   PolarityCheck();
767
768   break_time = millis() + DcTestDuration;
769   while (millis() <= break_time) {
770
771     float average = 0;
772
```

```
775       value += analogRead(A7);
776     }
777     value = value / 50;
778
779     DcVoltage = ConvertVoltage(value);
780
781
782     if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
783       previousMillis = millis();   // update time every second
784       lcd.setCursor(0, 3);
785       lcd.print(DcVoltage);
786       lcd.setCursor(5 , 3);
787       lcd.print(" Volts DC      ");
788     }
789     PolarityCheck();
790   }
791   LowVoltageSpec = DcVoltage;
792   digitalWrite(HighOhm, LOW);
793
794   digitalWrite(LowVoltage, LOW);
795   delay(100);
796   digitalWrite(LineVoltage, HIGH);
797   digitalWrite(LowOhm, HIGH);
798
799   break_time = millis() + DcTestDuration1;
800   while (millis() <= break_time) {
801     PolarityCheck();
802     if (DcVoltage < 11)
803     {
804       digitalWrite(AcOn, LOW);
805       digitalWrite(3, LOW);
806       delay(100);
```

```
    PolarityCheck();
  }
  float average = 0;


  for (int i = 0; i < 50 ; i++) {
    value += analogRead(A7);
  }
  value = value / 50;

  DcVoltage = ConvertVoltage(value);


  if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
    previousMillis = millis();   // update time every second
    lcd.setCursor(0, 3);
    lcd.print(DcVoltage);
    lcd.setCursor(5 , 3);
    lcd.print(" Volts DC      ");

  }
}
VoutFullLoad1 = DcVoltage;

break_time = millis() + FullLoadDuration;
while (millis() <= break_time) {
  PolarityCheck();
  if (DcVoltage < 11)
  {
    digitalWrite(AcOn, LOW);
    digitalWrite(3, LOW);
    digitalWrite(LowOhm, LOW);
    delay(100);
```

```
840       PolarityCheck();
841     }
842     float average = 0;
843
844
845     for (int i = 0; i < 50 ; i++) {
846       value += analogRead(A7);
847     }
848     value = value / 50;
849
850     DcVoltage = ConvertVoltage(value);
851
852
853     if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
854       previousMillis = millis();   // update time every second
855       lcd.setCursor(0, 3);
856       lcd.print(DcVoltage);
857       lcd.setCursor(5 , 3);
858       lcd.print(" Volts DC      ");
859
860     }
861   }
862   VoutFullLoad2 = DcVoltage;
863   LoadRegFlag = 0;
864   CalcFlag = 1;
865 }
866
867 void Calculations()
868 {
869   VoltageRegulation = (DcMax - VoutFullLoad2) / VoutFullLoad2 * 100;
870   VLR = (VoutHigh - VoutLow) / VoutHigh * 100;
871   lcd.clear();
```

```
  Serial.print(NominalVoltageSpec);
  Serial.print(",");

  Serial.print(LowVoltageSpec);
  Serial.print(",");

  Serial.print(VoutFullLoad1);
  Serial.print(",");

  Serial.print(VoutHigh);
  Serial.print(",");

  Serial.print(VoutLow);
  Serial.print(",");

  Serial.println(rippleVoltage);

  ExcelFlag = 0;

}//  Serial.println("LABEL,Computer Time,Student ID,Minimum DC Out
  lcd.setCursor(0, 0);
  lcd.print("Voltage Reg = ");
  lcd.print(VoltageRegulation, 3);
  lcd.setCursor(0, 2);
  lcd.print("VLR = ");
  lcd.print(VLR, 3);
  CalcFlag = 0;
  delay(10000);
  ExcelFlag = 1;
}

void SendToExcel()
{
  Serial.print("DATA,TIME,");

  for (int f = 0; f < 4; f++) {
    Serial.print(studentIdinput[f]);
  }
  Serial.print(",");

  Serial.print(DcMin);
  Serial.print(",");

  Serial.print(DcMax);
  Serial.print(",");

  Serial.print(VLR);
  Serial.print(",");

  Serial.print(VoltageRegulation);
  Serial.print(",");

  Serial.print(NominalVoltageSpec);
```

Code for Nano

```
int AC_pin = 11;//Pin to OptoTriac
volatile int dim = 0 ; //Initial brightness level from 0 to 255
int minus = 7;
int AcOn = 9;
int plus = 8;
int HighVoltage = 6;
int LineVoltage = 5;
int LowVoltage = 4;

void setup()
{
  Serial.begin(9600);
  pinMode(0, INPUT_PULLUP);
  pinMode(AC_pin, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(3), light, FALLING);//Whe
  pinMode(minus, INPUT);
  pinMode(AcOn, INPUT);
  pinMode(plus, INPUT);
  pinMode(HighVoltage, INPUT);
  pinMode(LineVoltage, INPUT);
  pinMode(LowVoltage, INPUT);
  dim = 0;
}

void light()
{

  if (dim < 1) {
    //Turn TRIAC completely OFF if dim is 0
    digitalWrite(AC_pin, LOW);
  }
```

```
  if (dim > 254) { //Turn TRIAC completely ON if dim is 255
    digitalWrite(AC_pin, HIGH);
  }


  if (dim > 0 && dim < 255) {
    //Dimming part, if dim is not 0 and not 255
    delayMicroseconds(34 * (255 - dim));
    digitalWrite(AC_pin, HIGH);
    delayMicroseconds(350);
    digitalWrite(AC_pin, LOW);
  }
}


void loop()
{
  Serial.println(dim);
  while (digitalRead(AcOn) == 0) {
    dim = 0;
    Serial.println(dim);
  }

  if (digitalRead(plus) == 1)
  {
    if (dim < 253)
    {
      dim = dim + 1;
      delay(25);
    }
  }

  if (digitalRead(minus) == 1)
```

```
    }
  }

  if (digitalRead(minus) == 1)
  {
    if (dim > 69)
    {
      dim = dim - 1;
      delay(25);
    }
  }

  if (digitalRead(HighVoltage) == 1)
  {
    dim = 255;
  }


  if (digitalRead(LineVoltage) == 1) {
    dim = 207;
  }

  if (digitalRead(LowVoltage) == 1) {
    dim = 186;
  }

}
```

Test Results

Test 1 – User Interface System Check

Test Description	This test will be used to check that the UI screen (LCD screen) outputs the correct display throughout the process.

Step	Action	Expected Results	Observed Results
1	Power Supply Test Unit is turned on.	LCD will indicate the user to connect the power supply to the tester.	LCD correctly instructs user to connect power supply and check polarity if incorrect.
2	Power supply is connected to the tester.	LCD will indicate that the power supply is connected and that the test mode can be selected.	LCD indicates that the test can be started.
3	Start button is pressed.	LCD will indicate that testing is in progress, along with the current power supply voltage.	Press any button to start is displayed. Pressing any button correctly runs the test.
4	Test completed.	LCD will indicate that the test is complete.	LCD indicates that the test is complete and can be run again by pressing any button.

Test State: Pass Passed ____ Failed

Tested by: Mitchell Date: 4-22-20

Comments: Additional functionality can be added if requested.

Test 2 – Check Relay and Resistor Selection

Test Description	This test will check to ensure proper functionality of the relay board and the code controlling the relay board.

Step	Action	Expected Results	Observations
1.	Start button is pressed and test is started.	Relay board will switch to correct resistor for the test being run.	Relays switch between 20 Ohm and 15 Ohm at the correct times.
2.	Test completed.	Relay board switches back open contacts for no load.	Relay board correctly switches back open contacts for no load when done.

Test State: PASS_ Passed ____ Failed

Tested by: ___Mitchell_____ Date: _____4/22/20_____

Comments:

Relays switch more than originally planned for to allow for better functionality. This is intended and allows for a better testing system by reducing power draw on the students power supply before the 5 minute test.

Test 3 – Ensure Correct Timing Requirements are Fulfilled

Test Description	This test will check to ensure proper functionality of the microcontroller timing code.

Step	Action	Expected Results	Observations
1.	Start button pressed.	Test begins and timer starts timing when 15 ohm load is applied.	Internal timer begins when the 15 Ohm load is connected.
2.	Five minutes of testing is reached.	Microcontroller stops test and records results to computer. Test duration time is also displayed on LCD.	Test lasts exactly 5 minutes. LCD does not currently display timer.

Test State: PASS_ Passed ____ Failed

Tested by: ___Mitchell_____ Date: ___4-22-20_____

Comments: LCD timer display is not a required function of the tester. It will be added on if time allows. Other core functionality was completed before the timer display.

Test 4 – Check for Correct Voltage Output from Tester.

Test Description	This test will check to ensure proper functionality of the AC voltage controller.

Step	Action	Expected Results	Observations
1.	Press start button and measure AC voltage at each section of test.	Power supply test unit should output 125, 117, or 110V RMS depending on the section of the test.	All voltage levels were correctly controlled within 100 mV. Required was within 1V.

Test State: PASSED_ Passed ____ Failed

Tested by: ___Mitchell_____ Date: ___4-22-20_____

Comments:
System is 10X more accurate than the minimum requirement.

Test 5 – Check Data Transmission

Test Description	This test will check to ensure proper functionality of the USB data connection from the microcontroller to the computer.

Step	Action	Expected Results	Observations
1.	Test is completed.	LCD displays results and a pass or fail condition.	Does not currently display pass or fail. Not a required section. Will be added if time allows.
2.	LCD displays results.	The results stored in the microcontroller are sent to a spreadsheet on the computer over USB.	Results saved to excel at end of test. File is autosaved as date, time, and ending the file name with the students given ID number.

Test State: _PASS_ Passed____ Failed

Tested by: ____Mitchell Brauns_____ Date: ___4-22-20_____

Comments: Saves data in under 1 second.

Test 6 – Check Test Unit's Response to Failures

Test Description	This test will check for proper operation under power supply failure conditions.

Step	Action	Expected Results	Observations
1.	Slowly lower DC voltage.	Test should stop and disconnect the load. LCD should display a failure diagnostic.	Stops test, cuts power, and indicates to check polarity.
2.	Raise voltage over 20V DC.	Test should stop and disconnect the load. LCD should display a failure diagnostic.	Unable to test this due to lack of equipment at home. Available power supply ranges from 1.2 to 12.8V.
3.	Turn off power supply during test.	Test should stop and disconnect the load. The power to the power supply should also be cut off. LCD should display a failure diagnostic.	Stops test, cuts power, and indicates to check polarity.
4.	Apply 3V RMS ripple voltage in place of power supply.	Test should stop and disconnect the load. The power to the power supply should also be cut off. LCD should display a failure diagnostic.	Unable to test due to lack of equipment. Need oscilloscope to calibrate ripple measurement and function generator to test functionality.

Test State: PASSED_ Passed ____ Failed

Tested by: Mitchell Brauns Date: 4-22-20

Comments: All functionality that was able to be tested from home was tested. Further testing can be done in the IUPUI lab once open.

Final Product Pictures

Recommendations

The project is fully functional as is, but there are some improvements that could be made in the future. The communication lines for controlling the AC voltage are set up and available to use. It will take some tweaking in the code and testing with a variac to get this set up. This was planned to be set up, but a variac is needed to get it fully functional. The base code to get it working is there, so it shouldn't be too difficult with the correct equipment. The tester will still automatically vary the voltage, but it is based on set points once it is calibrated for a location. Future improvement will bridge the setpoints, calibration, and auto-adjustment into one function. This is already partially set up.

The other recommendation would be to add a Zener diode to clamp the gate to source voltage of the P channel MOSFET. This will keep the tester safe to the desired 50V, it is currently safe to 25V. This add on must be done since a Zener diode was destroyed during testing, and no replacement was available in time due to COVID-19. Otherwise the pandemic didn't affect our project too much.

References

Kumar, P. (2019, July 12). Arduino Based AC Voltage Control using Zero Voltage Crossing Detection. Retrieved from https://www.engineersgarage.com/contributions/arduino-based-ac-voltage-control-using-zero-voltage-crossing-detection/.

Measure Any AC Voltage Up to 250V. (n.d.). Retrieved from https://create.arduino.cc/projecthub/SurtrTech/measure-any-ac-voltage-up-to-250v-1af7bd.

Reverse Polarity Protection. (2018, August 27). Retrieved December 2, 2019, from https://circuitdigest.com/electronic-circuits/reverse-polarity-protection-circuit-diagram.

Abubakar, I., Khalid, S. N., & Shareef, H. (2017, February). CALIBRATION OF ZMPT101B VOLTAGE SENSOR MODULE USING POLYNOMIAL REGRESSION FOR ACCURATE LOAD MONITORING . Retrieved from http://www.arpnjournals.org/jeas/research_papers/rp_2017/jeas_0217_5728.pdf

Shahbaaz. (1968, January 1). Retrieved from https://www.extendoffice.com/documents/excel/4620-excel-autosave-after-entry.html

(2018, August 27). Retrieved from https://circuitdigest.com/electronic-circuits/reverse-polarity-protection-circuit-diagram

Document Approval

This section includes signatures of those that have written, reviewed or approved this Requirement Definition validation deliverable for the Power Supply Tester Automation Senior Design Project.

Authors' Signatures:

Your signature indicates that this document describes the Requirements of the Project Name Here and that this deliverable meets IUPUI standards for documentation.

Name	Signature	Title/Department	Date
Yazed Alanazi	Y. Alanazi	Author	11/5/19
Mitchell Brauns	Z Mitchell Brauns	Author	11/5/19

Approver's Signature:

Your signature signifies that you agree with the purpose and scope of this document, and that it has been reviewed by appropriate personnel to ensure compliance IUPUI policies.